COTS: Is it Just a ✔ for Your Program?

Or Are You a Real Part of Acquisition Reform ?

LUKE CAMPBELL

The subject of Commercial Off-the-Shelf (COTS) is complex because there is no single COTS issue — there are many, depending on your perspective and position in the acquisition life cycle. In addition, the overall picture remains clouded by wild speculations about COTS savings and advantages, which are at least partially true. Any new concept requires some amount of hype to establish a critical mass to get it underway. We are beyond that stage with COTS, and the facts must now emerge. In this article, I examine COTS effects on three broad life cycle phases: development, support, and future growth or upgrades.

A Little Background

COTS is often touted as "we-do-it." However, if a program truly embraces COTS, it becomes apparent that while *some* acquisition changes are required during development — *substantial* changes are required during support.

To many, COTS is synonymous with computers. Most "computer experts" have only minimal understanding of COTS and base their acquisition goals on a wholly insignificant view of the life cycle. To be blunt, just because you have a computer on your desk, does not make you an expert on the subject of COTS use. The primary differences between COTS in desktop systems and COTS in weapon systems are desktop integration vs. platform integration and life cycle times. Big differences. More will be discussed on these later.

You might say there are two COTS philosophies: Little COTS and Big COTS. Little COTS philosophy says, "We looked at commercial systems," or "We use an Intel processor." The information in this paper is based on Big COTS philosophy regarding the E-2C aircraft and its Mission Computer Upgrade (MCU). A sampling of Big COTS in the MCU is as follows:

- The operator-display workstations are developed from a performance-based specification, which is not under the control of the PMA.
- MCU runs a UNIX Operating System (OS).
- MCU uses operational software in C++, which is developed by a University.
- MCU is connected by Ethernet to a mission computer repackaged to fit in the existing volume, but with no design changes from its commercial counterpart.
- MCU also runs a second commercial UNIX OS.
- COTS Cooperative Engagement Capability hardware and software are connected by a second Ethernet connection.

The Hawkeye 2000 Airborne Early Warning and Control aircraft being test flown by the U.S. Navy.

All these highly volatile systems must play in tune. This is more than just a considerable configuration management challenge: a methodology must also be in place for Technology Insertion (TI). This is Big COTS.

The concept of, "Just insert new technology during production," ultimately became one of the most questionable strategies used to initially support the COTS philosophy. It implied two major considerations that were, apparently given little thought at the time:

***Campbell** is the PSA IPT lead, Naval Air Warfare Center, Patuxent River, Md.*

Photos courtesy Northrop Grumman

- COTS didn't need to be tested.
- Somehow, money would be appropriated for this condition.

Attempting to cross-dress a well-known concept in a simple-office PC environment to a complex weapon system with an even more complex acquisition cycle—the support process—can lead to disaster.

Disaster? What Disaster?

The E-2C operator workstations were developed by a sister Systems Command (SYSCOM) under a performance-based specification. Many units were not functional when installed into our system. Simple solution, right? Just get the vendor to fix what's broke. The problem was that the units weren't broke. Since the delivered products all passed contractor-factory acceptance tests, there was nothing for the vendor to fix. Perplexed? So were we. One of our first *lessons learned* was that Class II changes to the vendor became Class I problems in our system. After considerable discussion about this situation, the vendor was very amenable. Establishing many Process Action Teams, over a period of one year the vendor made sweeping analysis of sub-vendors and developed very detailed processes—but the results of the product did not change. Problems such as this occur several times each year, and program managers electing to use COTS must be prepared to solve them.

Our solution is a Program Support Activity (PSA). The PSA subsumes the clas-

The concept of, "Just insert new technology during production," ultimately became one of the most questionable strategies used to initially support the COTS philosophy.

This is the business end of the Northrop Grumman E-2C Hawkeye 2000 Airborne Early Warning and Control aircraft: the operators' consoles. Taking advantage of commercial off-the-shelf technology and commonality with workstations used on U.S. Navy ships, these workstations are just one of the several significant system upgrades constituting the Hawkeye 2000 system. The heart of Hawkeye 2000 is the mission computer upgrade.

sical Software Support Activity (SSA) functions, but adds the critical functions of Technology Insertion, a clearinghouse for Configuration Management, and what we call color-coding. More will be discussed on this later.

A Bumpy Start — A COTS Failure

We tried to use as much COTS as possible. One of the first things we tried was to use the COTS databases. We found that they were big — very big and S-L-O-W — glacial, to be exact. The basic problem with commercial databases is that "real time" to them is similar to a transaction at an Automated Teller Machine (ATM). During the same time it takes for that "fast cash" ATM transaction, several enemy fighters need to be shot down. "Tactical" to these commercial databases means to get the card out of the ATM because it's dark, and the person coming up behind you is unknown. For DoD, several missiles impacted a destroyer in this same period. "Oops."

The *lesson learned* was perspective. Our military needs were not only well in advance of our commercial needs, but more disturbing, were also in advance of industry understanding of the concept of speed and performance. The problem was that our market is quite small — insignificant — in fact. Military systems are an oddity to industry — a speck of dust.

We did not give up on commercial databases; we discussed performance with the vendors at some length. After discussions with industry about speeding up their databases, *our solution* was to go back to the "do-it-yourself" database, and give up on COTS *this time*. Another *lesson learned* was that decent system engineering and analysis of industry products is necessary. So called *vaporware*, or software that is seemingly never delivered, is rampant. *Our solution* was to take a six-month loss in schedule.

More Vaporware

We also wanted a multi-level-secure environment. After performing surveys for capability and market share, we chose Digital Equipment Corporation's (DEC) Multi-Level Security+ (MLS+) system. This lasted for four years before DEC announced that the market for this product was not nearly as strong as envisioned, and the product would be discontinued. The *lesson learned* was when you use COTS, be prepared for change. Industry moves to the beat of quarterly profit — period. Fortunately for us, we were not entirely unprepared for this eventuality, and *our solution* was to fall back on plain-old UNIX, and use our well-designed software architecture for the security features we need.

COTS Computer Performance — Some Perspectives

The performance growth curve for military computer systems has been virtually flat for the past 10 years because of COTS use. We are only now climbing the curve again. Blasphemy? Let's look at the data. Certainly, there is no argument that the raw power of hardware is light years faster than it was 10 years ago, and there are no 640K memory barriers. But consider the system. Think about the desktop applications you run today and the performance of those applications 10 years ago.

On the negative side, your disk drive is *still* 90 percent full — except that today it's 2 gigabytes, while it was 20 megabytes back then. True, you didn't have 100 megabytes of "essential" pictures from the World Wide Web. Or consider word processing. The file size of a page of text — just plain text — is 30K, compared to 2K back then. What about performance? Do you actually see the 366 megahertz speed of the latest Pentium compared to the 2 megahertz Z-80? Certainly systems are faster — but 150 times faster? Efficiency is no longer a part of our vocabulary.

So What, You Ask? Let's Look at the Positive Side

Today, we can easily embed pictures in documents, making them highly readable and understandable. We can ship them around the world at breakneck speeds (assuming the network is up today). We can develop huge spreadsheets for Team Work Plans and Earned Value Management. Who doesn't like having the ability to make a presentation in color, with pictures, sound, animation, and 10 or more fonts using an electronic projector? Is it even *possible* to still make a presentation with short bullets on a typed sheet, which are copied to a transparency for use on an overhead projector? On another front, new Computer-Aided Design (CAD) software applications speed up designs, basically eliminate paperwork, and perform automatic calculations, saving untold workhours and millions of calculation errors.

While much of this is tongue-in-cheek, the point is that while we have applications that provide massive capabilities; for the most part, this same software has gobbled up the hardware performance gains in the past 10 years. Software is the Achilles' heel of COTS use — Operating Systems that used to take 5K worth of memory in the "write-it-yourself" good old days now consume megabytes with a commensurate use of processor throughput. Compilers that produced highly optimized code now define optimization in megabytes; but that's okay, since we have gigabytes of memory, and to vendors, memory is like the nacho chip commercial with Jay Leno — "Just eat them, we'll make more."

What's the Point?

Bettering software efficiency means the same hardware can perform more functions. Conversely, bettering efficiency means we don't need as much of that same hardware — saving weight, power, volume, cost, and Mean Time Between Failures (MTBF). *A full frontal assault from the military to private industry in the area of software efficiency could reap huge benefits*. Frankly, this would also benefit industry for most, but not all, of the same reasons. "Hmmm."

Development Phase — Good News

So those were a few major issues during development. The COTS *good news* is that other than the trivial cost of buying a license for use, we did not have to develop an Operating System. We did not have to *develop* a computer. We did not have to *develop* Input/Output and protocols. This all translated to dollars. The new COTS mission computer costs

about $2.5 million less per unit than the existing mission computer (which is no longer supported by the manufacturer). On a packaging COTS-O-Meter scale ranging from commercial to mil-qualified, and since no data existed, we were leery of off-the-shelf. Hence, we paid $12 million to have the new mission computer repackaged and fully qualified for our use. To date, we have experienced no packaging failures from our mission computer.

At the same time, the operator workstations were developed by our sister SYSCOM for shipboard use. We paid nothing for this development. Due to our association with a larger shipboard market, our unit cost for these workstations dropped by over 60 percent in a couple of years. On the other hand, their approach to packaging was less stringent than our mission computer—and it showed. We have experienced failures of connectors and pins; and on one assembly, use of stock that was too thin has caused warping and physical failure. Lack of space on the boards for marking is a problem. However, under the contract the manufacturer is taking all this in stride, and together with our sister SYSCOM, these problems are being addressed and corrected.

Second Phase—Support

The basic problem during support can be summed up by the Class I/Class II scenario mentioned earlier. For our workstations, we have no guarantee that yearly installs, spares, or repair-buys will procure the same functionality. A recent example is sub-vendor firmware change in the keyboard. The vendor changed diagnostics and the OS to accommodate this Class II change, and units passed the vendor's FAT (Factory Acceptance Test). However, when installed in the E-2C, the units did not work. OS calls had to be modified in the software. While this was a relatively minor difficulty that cost a couple of weeks and the minor sum of tens of thousands of dollars, it pointed out a future discrepancy. Our *lesson learned* was that parts—such as spares—leaving the vendor would have to be screened before entering supply, lest we fill the pipeline with scrap.

We are still grappling with this problem today. Support is historically based on Aircraft Procurement Navy-5 (APN-5) dollars after deployment. However, with COTS upgrades occur, which can change end-system functionality. An argument can be made by Naval Inventory Control Point (NAVICP) that the spares and repairs they purchased were exactly the part number they were given—so it's not on their watch. Conversely, the program office argues that the development is over, integration is complete and it works—so it's not on our watch. "Hmmm."

Funding has been a point of difficulty for some time. Our solution to this conundrum has been to request [and so far, successfully defend] an APN-5 Operational Safety Improvement Program (OSIP) that allows us to test these new technology installs, spares, and repairs before they are used in our aircraft. This is done very economically using the same bench assets we have for long-term software support. It does, however, stick an integration agency squarely in the middle of the supply pipeline—leaving logisticians squirming even more than usual.

We need the DoD community to understand that we are not an isolated case—more will follow. The traditional boundaries, which define colors of money, are being drug across each other.

Third Phase—The Future Technology Insertion and the Great Unknown

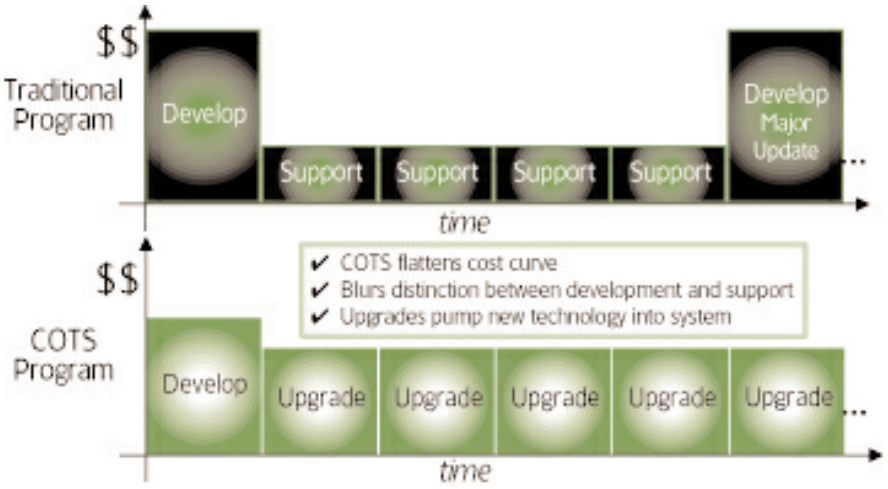
Technology Insertion comes about for three primary reasons: yearly buys, Plug 'N Play, and future upgrades. The first occurs when you attempt to make yearly buys of COTS equipment. Just like the PC market where you cannot buy last year's model, and you have to accept the faster processor, more memory, and bigger disk drive (usually, at a lower price), likewise, our COTS equipment changes too. For our mission computer, we have roughly a four-year cycle. For our workstations, the cycle is about every year. To mitigate the latter, our solution was to block-buy two years' worth of Government Furnished Equipment (GFE) at the same time.

There is a cost associated with these changes, as well as time to test. They cannot be blindly inserted. Operating Systems, firmware, protocols, and diagnostics change—and these changes generate changes to the operational software. The Operational Evaluation community has not been hit by this creeping technology yet—but they will. In our case, we are now planning the 2001 aircraft to have next-generation workstations, while the 2003 aircraft will have next-generation mission computers. Our approach to these upgrades is that they do not extend the functional capability from a Fleet point-of-view, nor will they change the repair scenarios. As such, while we contemplate the test suite for these upgrades, we do not intend to have Operational Evaluations. *This is another area where a common point-of-view and approach across DoD would be most beneficial.*

A second technology-insertion cause is Plug 'N Play. In our case, while we do not have the selection of software en-

Military needs were not only well in advance of our commercial needs, but more disturbing, were also in advance of industry understanding of the concept of speed and performance.

Impact of COTS



joyed by the games community at the local electronics superstore, there are software capabilities such as Air Task Order and fusion algorithms which have been developed by other organizations. This software requires some amount of integration, but its insertion time is considerably less than a start-from-scratch development — with commensurately smaller cost.

The third technology-insertion cause is future upgrades. A major radar upgrade for the E-2C looms on the near horizon. This will require considerably greater computer performance than we have now installed. The thought for many years is that we would take advantage of the COTS performance increases through time and install new computers commensurate with the production of new radar systems.

The point of using COTS is to avoid the large development costs historically associated with new upgrades. The cost of this large avoidance will be a continuum of smaller costs between development and upgrade (opposite chart).

Benediction

Actual data, and therefore, concrete answers for the full life cycle are not yet available, and in a rapidly changing organization, they may not be of value for long. It's hoped that the experiences outlined here may best help by stimulating thinking for additional solutions and discussion.

To date, we have saved money and provided the fleet with capability through COTS — and we're not done.

One point is clear, we need understanding and flexibility regarding the total life cycle of COTS, and we don't have years to achieve this end-game. We need changes in acquisition to save more money to continue program success.

If understanding and flexibility are not achieved, COTS will become just another ☑. We have too many of these now.

Editor's Note: The author welcomes questions or comments on this article. Contact him at **CampbellLO@navair.navy.mil.**

Foreign Students, Dignitaries from Japan Tour DSMC Main Campus

Students and staff from the Graduate School of Security Studies, National Defense Academy (NDA), Japan, tour the DSMC main campus, Fort Belvoir, Va., March 6 to improve their understanding of U.S. armed forces and their acquisition organizations. This is their third year to visit, and DSMC has continued to promote and encourage this constructive engagement and interchange with our allies. Pictured from left: Maj. Akio Tomita, Japanese Ground Self Defense Force, Research Associate of the Research Committee; Lt. Cmdr. Yasufumi Miyahara, Japanese Maritime Self Defense Force, NDA student; Lt. Gen. (Ret.) Naruhiko Ueda, Senior Executive Director, Defense Research Center; Air Force Brig. Gen. Frank J. Anderson Jr., DSMC Commandant; Lt. Minako Hayashi, Japanese Ground Self Defense Force, NDA student; Capt. Takeshi Yanagitani, Japanese Air Self Defense Force, NDA student; Huniichi Tanida, NDA student; Tony Kausal, DSMC Air Force Chair.

Photo by Richard Mattox